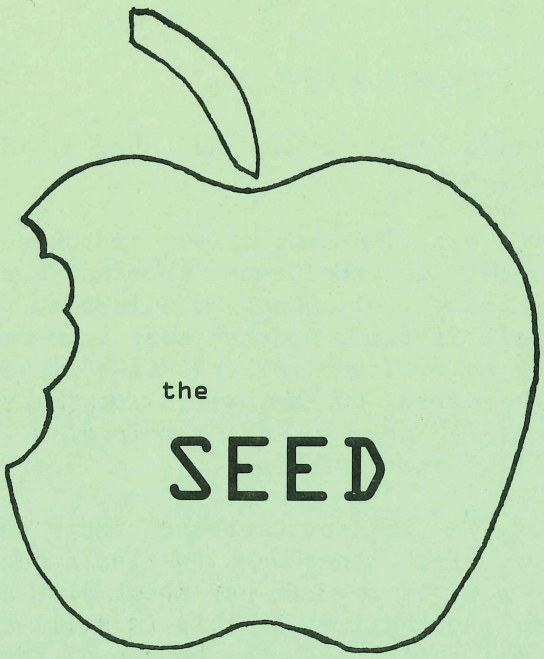


OFFICERS

PRESIDENT
Steve Martin (303) 469-6855
VICE PRESIDENT
Dave Eisler (303)694-6217
TREASURER
Fred Yates (303)988-9053
SECRETARY
Dave Moger --
LIBRARIAN
Bob Gaiser (303)779-4033
SEED EDITORS
Frank Reeves (303)499-2192
Louis Breyfogle (303)449-6454
Ron Abbott (303)772-7198
OFFICERS-AT-LARGE
Dan Geisler (303)832-6465
Al Tara (303)755-8009

Volume 4 Number 7 November 1982 Price $1.50

MEMBERSHIP INFORMATION
$10 APPLICATION FEE AND $18 PER YEAR DUES

Seed Editors
3015A Bluff St.
Boulder, CO 80301

FIRST CLASS MAIL
U.S. POSTAGE
PAID
Permit No. 502
Boulder CO 80302

PRINTED BY KWIK KOPY

GENERAL MEETING SCHEDULE
(subject to change !!)

NOV - Demonstration of new Apple II plug in card by ALF Products Inc. *

DEC - Demonstration of the new "SAVEY CARD" personal language system by Scott Knaster *

JAN - CPM demonstration by Bob Gaiser

FEB - Business Applications *

MAR - Assembly Language *

APR - Home Control

MAY - Election of Officers

* = revised since last month

TREASURER'S REPORT

October Treasurer's Report		
Sept. Ending Bal.		2,944.32
New Membership		710.00
Renewals		301.50
D.O.M. Sales		280.00
Club Library Copying		158.00
Advertising		.00
Sales of Supplies		1,687.03
Interest Received		8.79
Misc.		.00
Supplies for:		
President	<	.00>
Vice President	<	.00>
Secretary	<	65.00>
Treasurer	<	7.00>
Librarian	<	540.96>
Editor	<	.00>
Officers at large	<	36.55>
PURCHASE FOR RESALE	<	1,411.00>
Balance as of 10/12/82		4,029.13

Respectfully submitted
Fred Yates -- Treasurer
October, 1982

FROM OUR PRESIDENT

Well there's good news and there's bad news this month.

First the bad. As many of you are aware we have met in the Green Center at the Colorado School of Mines in Golden ever since Apple Pi first started over 4 years ago. The main reason for this is that we have had access to the auditorium there at no charge thanks to the help of Dr. Brown at the School of Mines.

Unfortunately it now appears that in January we will either have to find a new place to meet or we will pay about $100 a month for the facilities. This is still a bargain as it will include the use of the Advent television which usually costs $100 alone for a night. However we have found a way that we might be able to pay for the use of the room and generate some publicity for the group. What we are going to try is a Raffle. The prizes will probably be a Z-80 card and a good software package. Full news will be available at the next meeting.

Now for the good news. We kicked off the beginings of the Special Interest Groups last month at the regular meeting and the interest seems to be good. I have another article elsewhere in this issue for those people who would like to get into one of these groups.

The president now has a Modem which will allow for exchange of articles and software for those who have modems in the club. I have also set up a Bulletin Board on a System called UFONET. To access the board call 278-4244 and after giving your name and city you can access the Apple Pi board through the "Gateway" option. You may leave messages to officers or other members here and I will leave club news on the board. An upload and download feature will be added soon so we can form a software exchange. Public Domain only please.

The November General meeting will focus on new products from ALF. They will show off there 8088 card for the APPLE with some new software.

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$                                      $$
$$   :::---<<< DENVER'S >>>---:::      $$
$$                                      $$
$$      APPLE PI USER'S GROUP           $$
$$      ====================            $$
$$        is giving away  2             $$
$$                                      $$
$$    $50.00  Gift Certificates         $$
$$                                      $$
$$      good at a store that            $$
$$    is a member of APPLE PI           $$
$$                                      $$
$$     1.  For the best Program         $$
$$  written and given to APPLE PI       $$
$$  in any Computer Language that       $$
$$    runs on an APPLE Computer         $$
$$                                      $$
$$ 2.  For the best HELLO GRAPHICS      $$
$$          program for our             $$
$$        DISK OF THE MONTH             $$
$$                                      $$
$$   DEADLINE:  December 31, 1982       $$
$$                                      $$
$$   MUST BE A MEMBER OF APPLE PI       $$
$$                                      $$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

The December General meeting will have a demonstration of the new SAVVY card which has been called an artificial intuitiveness operating system. We also plan to have plenty of door prizes for a Christmas special.

As usual the officers will have a board meeting the second Thursday in November at the Green Center in Room 274. All members are invited.

Steven Martin
October 21, 1982

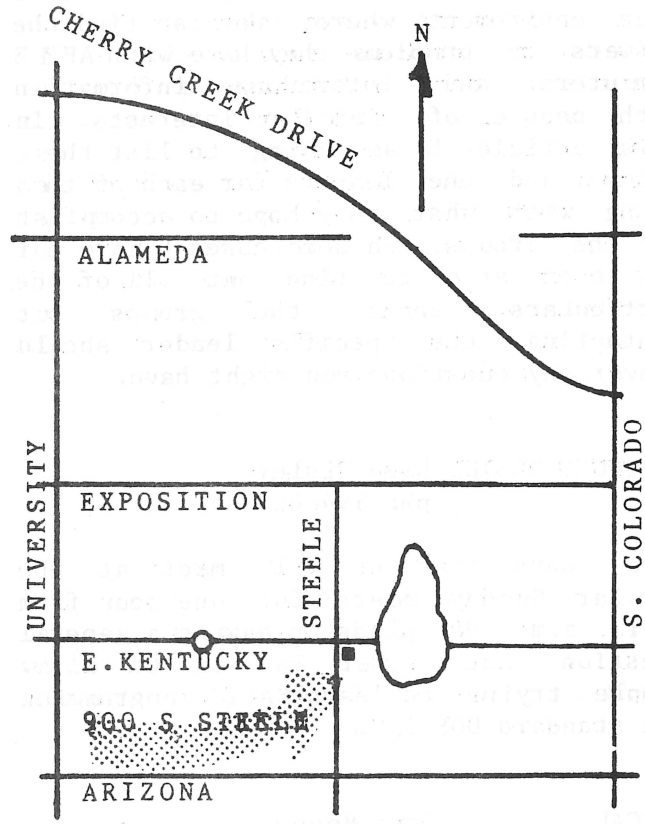
Apple Pi Benefits

1. Program library
2. Club buying power on supplies
3. Publication library
4. Monthly meetings
5. Newsletter
6. Member Forum

FDR

NOVEMBER Swap Meet

Our November Swap Meet will be held on Sunday November 7, from 10AM to 3PM at 900 S. Steele Street in Denver. Our host is Dr. D. E. Harvey, who may be reached at 777-6317 for instructions if you get lost.



November GENERAL MEETING

Our next regular meeting will be held at:

COLORADO SCHOOL OF MINES
GREEN CENTER
GOLDEN, COLORADO

7PM November 4, 1982

We hope to see you there
Meetings are monthly scheduled for the first Thursday of each month.

SPECIAL INTEREST GROUPS

by Stephen Martin

At the September General meeting we kicked off a set of Special Interest Groups geared at providing a means of getting members together in a face to face enviroment where they can find the answers to problems they have with APPLE computers and interchange information with people of simialar interests. In this article I am going to list these groups and the leaders for each of them along with what they hope to accomplish in the groups. In some cases I have not yet been able to find out all of the particulars about the groups but contacting the specific leader should answer any questions you might have.

ADVANCED BASIC: Dave Eisler
ph: 694-6217

Dave says that he will meet at the regular Sunday meets for one hour from 10-11 a.m. He plans on having a general Question and Answer Session to allow people trying to learn BASIC programming and standard DOS I/O.

PASCAL: Dave Moger
ph: ?

Dave is still in the process of getting settled into Colorado and does not yet have a phone. Dave has considerable experience with the XPLO language and other operating systems. While he claims not to be an expert at PASCAL he should be able to answer most questions. From all appearances this may be the largest interest group at the start. Dave plans to meet at his house or a members house on a monthly basis. Contact Dave at the General meetings, he's also our Secretary.

ASSEMBLY LANGUAGE: Bryan Johnson
ph: 444-2476

Bryan is from Boulder and while he also claims not to be an expert on the APPLE he has a very good knowledge of computers in general and will have a good group from first indications. Bryan plans to meet at members houses.

EDUCATION: Dr. Austin Brown
ph: 279-5388

Dr. Brown has been an educator at the Colorado School of Mines in the Computer Center for more years than he may reveal. He was one of the founders of the APPLE Pi group and a long time user of the APPLE computer.

CP/M: Bob Gaiser
ph: 779-4033

Bob's group will meet at the begining of the Sunday meetings. Bob boasts a large library of public domain CP/M software and has used this operating system for some time now. Naturally, to be a member of the group you should own a Z-80 card. Bob is also our Librarian.

BUSINESS: Mike Gibson
ph: 666-9771

The business group was formed at the last minute but Mike plans to have meetings at the Sunday meetings.

We hope to be recieving regular updates on these groups from the leaders which will be published in the SEED. If you would like to join a group or would like to form your own group not on the above list please contact Dave Eisler or myself at the phone numbers on the front of the SEED.

SM

SEED ADVERTISING RATES

$20 ¼ Page

$35 ½ Page

$60 Full Page

God and the EPA

United States Representative Andrew J. Hinshaw of California inserted the following in the Congressional record of October 10, 1974:

In the beginning God created heaven and earth.

He was then faced with a class action lawsuit for failing to file an environmental impact statement with HEPA (Heavenly Environmental Protection Agency), an angelically staffed agency dedicated to keeping the Universe pollution free.

God was granted a temporary permit for the heavenly portion of the project, but was issued a cease and desist order on the earthly part, pending further investigation by HEPA.

Upon completion of his construction permit application and environmental impact statement, God appeared before the HEPA Council to answer questions.

When asked why he began the projects in the first place, he simply replied that he liked to be creative.

This was not considered adequate reasoning and he would be required to substantiate this further.

HEPA was unable to see any practical use for earth since "the earth was void and empty and darkness was upon the face of the deep."

Then God said: "Let there be light."

He should never have brought up this point since one member of the Council was active in the Sierrangel Club and immediately protested, asking "how was the light to be made? Would there be strip mining? What about thermal pollution? Air pollution?" God explained the light would come from a huge ball of fire.

Nobody on the Council really understood this, but it was provisionally accepted assuming (1) there would be no smog or smoke resulting from the ball of fire, (2) a separate burning permit would be required, and (3) since continuous light would be a wast of energy it should be dark at least one-half of the time.

So God agreed to divide light and darkness and he would call the light Day, and the darkness Night. (The Council expressed no interest with in-house semantics.)

When asked how the earth would be covered, God said, "let there be firmament made amidst the waters; and let it divide the waters from the waters."

One ecologically radical Council member accused him of double talk, but the Council tabled action since God would be required first to file for a permit from the ABLM (Angelic Bureau of Land Management), and further would be required to obtain water permits from appropriate agencies involved.

The Council asked if there would be only water and firmament and God said "let the earth bring forth the green herb, and such as may seed," and fruit free yielding fruit after its kind, which may have seen itself upon the earth.

The Council agreed, as long as native seed would be used.

About future development God also said: Let the waters bring forth the creeping creature having life, and the fowl that may fly over the earth."

Here again, the Council took to formal action since this would require approval of the Game and Fish Commission coordinated with the Heavenly Wildlife Federation and Audobongelic Society.

It appeared everything was in order until God stated he wanted to complete the project in six days.

At this time he was advised by the Council that his timing was completely out of the question. . .HEPA would require a minimum of 180 days to review the application and environmental impact statement, then there would be the public hearings.

It would take 10 to 12 months before a permit could be granted.

God said, "To Hell with it!"

ARTICLE FORMAT

For the moment, we would appreciate your submitting your articles in machine readable form. This implies that you give us a disk with one of the following on it:

3.3 DOS TEXT FILE (Apple Writer][)
3.3 DOS BINARY FILE (Apple Writer I)
APPLE PASCAL EDITOR FILE
3.3 DOS MAGIC WINDOW BINARY FILE

Your disk will be returned to you promptly. Be sure that you give us a label telling what you are giving us.

We are printing much of this issue with a Diablo using a Prestige Elite 12 pitch daisywheel, a column width of 3.5 inches (42 characters), and 56 lines per column. It is our intent to print program listings in 80 character widths. Diagrams and sketches will be dealt with on an individual basis - they do not necessarily have to be furnished camera ready, but we do need to be able to understand them!

An Incomplete Comparison of JRT and UCSD Pascal

by Ron Abbott

You may have noticed lately that serveral magazines have advertised JRT Pascal for $29.50. After reading several reviews, I ordered this p-code compiler.

This version, really a superset of ISO Pascal, runs under the C/PM operating system. Because of this requirement one must have a Z80 card of some sort, and the C/PM operating system installed. Another requirement is a 80 col card or terminal, Pascal is an 80 col oriented language.

JRT Pascal is a p-code compiler like UCSD Pascal (Apple Pascal) and M Pascal. There are two basic types of Pascal compilers available, one is a p-code compiler the other is a n-code compiler. A compiler is a computer program that translates ordinary text (ASCII characters) into a set of instructions which a computer can execute. In the case of a n-code or native code compiler the text is translated into machine code instructions directly executable by the C.P.U.

A p-code or puesdo code compiler interprets text into p-code or puesdo code. This p-code is not directly executable by the CPU, but rather requires a program to further translate this p-code into machine language instruction. Such a program is called a run time interpreter, and this program must be in the memory of the computer during the execution of the program. Why use a p-code program instead of n-code program? The p-code program has several advantages over a n-code version. Your program is very portable from computer to computer regardless of the CPU. All that is required is a custom written run time interpreter which is relatively easy to write compared to an entire n-code compiler. Because a language doesn't have to be customized for every different machine the amount of effort needed is greatly reduced.

Generally n-code compilers are much faster, more expensive (not always) and have little or no portability between dissimilar CPUs. Although slower than it's n-code cousin, p-code compilers produce programs that are quite compact and run 2-5 times faster than interpreted BASICS. To me the ideal development system would be a straight interpeter for program development and testing, and a native code compiler to squeeze all the speed out of the the program that is possible.

The process of producing any application program using a compiler goes something like this: Using a text editor or word processor, create a text file version of your program (called the source file). Then run the compiler, it uses your source file to produce an executable program or intermediate program. Edit the errors found by the compiler, which prevents a sucessful compiler, recompile. Repeat these steps until the program is executable. Run your compiled program using test data to determine if it is logically correct. If this process sounds laborious, you're quite right. Straight interpeted languages like Applesoft run slower generally, but during the development stage they are much easier to use. The old saying is "you don't get something for nothing" is very true.

Enough on background, back to JRT Pascal. This version of Pascal is a honest to God full implementation plus. It has features you would expect to find in compilers costing 10 times as much. In fact JRT used to be priced at 295.00. You can speculate on the reasoning for the price reduction youself. JRT comes with 150 page loose leaf style manual that is very complete. It is well organized, has many examples, and is reltively easy to read, but perhaps a little terse. It is not a tutorial, but rather a good reference, so you know Pascal or be perpared to learn from other sources. One complaint about the manual is the lack of an index, which I seem to constantly use.

There are several reviews written lately on JRT Pascal i.e. Creative Computing, Interface Age and Info World. So a straight review of this implementation

would be redundant. I thought a comparison between UCSD Pascal and JRT would be more interesting and perhaps relevent.

Comparing the data types shows the two to be quite similar to other implementations of Pascal. UCSD stores reals as 7 digits, JRT stores them as 14 digits coded bcd, so no lose of accuracy is possible. UCSD Pascal uses long integers to get around this limitation of their reals, JRT doesn't have or need long integer types.

String handling facilities in UCSD and JRT are almost the same, which is to yes quite good. JRT allows the '+' symbol to concatenate strings as well as the usual form of 's1:=concat(s1,s2)', which is more cumbersome. Some versions of Pascal require you to define strings as fixed length array of characters, which is very tedious to use. Both JRT & UCSD define dynamic strings, that is strings whose length is not predeclared.

JRT has an interesting builtin function that allows access to the address of a varible. UCSD doesn't allow direct access to to ram memory. Some other builtins unique to JRT is 'portin' and 'portout', these functions allow hardware oriented I/O. But alas, I'm afraid these won't work on the Apple because all I/O on a Apple is memory mapped, not port I/O. JRT has an 'upcase' function which converts a string to uppercase characters. This seems handy to me when comparing input from the keyboard and filtering it, you can test for the uppercase char only and not to test for upper & lower case characters. There's a 'hex$' function for converting a varible to it's hex format. UCSD has none of these functions or anything close, of course you could write them and install the functions in the system library. That one of the beauties of Pascal, being able to define new functions and procedures, which can be used over and over.

UCSD Pascal has 2 procedures for recovering dynamic pointer memory locations that are no longer being used, these are 'mark' and 'release'. JRT uses a standard Pascal procedure called 'dispose' which is more flexible. You can make direct calls to the C/PM operating system from a JRT program with 'call', UCSD has no equivilent. It is perhaps a philosophical difference between the two versions regarding direct access to ram memory. JRT offers several builtin functions and procedures which provide direct and indirect access to memory. UCSD, on the other hand, doesn't allow access except by variant records, but this I consider to be "unclean" and somewhat confusing.

Both support sequential and random file processing, UCSD uses the standard 'get' and 'put' procedures, JRT uses 'read,readln', 'write,writeln' procedures to acess files. While the JRT approach is non-standard and therefore less than portable, I consider it a clear and simple solution.

UCSD and JRT comes with a linker that can be evoked manually or automatically at compile time. A linker is a program that takes 2 or more pieces of a program a "links" them together to form one module. Large programs can, and are often, broken down to smaller segments during development. When each segment is compleated it can be linked to the main program before executing. An assembler is provided with each version, but I've not had a reason to use either at this time.

When is comes time to debugg your program JRT offers tracing procedures that follow the program flow and can indicate the line number and procedure/function name that was executing when a runtime error was detected. UCSD may be better on that score because this function is automatic, not so with JRT. JRT has a 'activan' procedure that monitors your program, while running, and outputs the number of times a procedure or function is executed within your program. Seems to me this would be very handy to optimize the program for max speed and efficiency.

JRT also offers, what appears to be some useful utilities with version 2.0. One is a graphing function program called 'jgraf', it produces a graph uses ASCII characters. 'Jstat' is a statistical

function which returns mean, std dev, variance, skewness, kurtosis and the first four moments about the mean. 'Letters' is a procedure which will output large block size headings on your printer (in ASCII letters), handy for labeling a report or manual.

The Nov 82 issue of "Byte" magazine contains 'The Users Column' a monthly column, which this month contains a benchmark program in Pascal used to evalute M'MT+, and several Basics. These benchmarks were run on a 8085 Compupro running at 6 mhz. I typed in the program verbatium and it ran with no changes in UCSD and JRT Pascals. (see benchmark listing)

EXECUTION TIMES IN MIN:SEC

UCSD Pascal	1:52
JRT Pascal	2:18

In the article many of the compiled Pascals and BASICs came out faster than these times, but they were running on a faster processor and they were native code compilers, not p-code compilers. I would consider the times for JRT and UCSD to be quite respectable.

Do I like JRT Pascal? You bet!! Obviously it's a steal for 29.50, they didn't even charge shipping and handling! Do I prefer it to UCSD? No, not yet, but I may in time. UCSD is a completely integrated system comprised of an editor (a very good one), filer, compiler and run-time intepreter. This integrated package makes program development about as easy as it can get, using a compiled language. JRT doesn't have an integrated editor so you must provide your own. I used Wordstar, which I really don't care for, and I don't think it's as good as the UCSD version. When developing a program one tends to jump back and forth between the compiler and the editor, correcting mistakes. It is always tedious and time consuming, but it's worse with JRT. On finding a error with JRT, execute Wordstar or eqivalent, give file name to edit, load the file, find the error, change offending syntax, exit Wordstar, execute JRT compiler. In UCSD compile errors require one keystroke to load the text file (for correction) and the cursor is resting on the offending statement. I didn't realize how convienent this was, until I had to use something different.

JRT Pascal will be my choice using C/PM operating system, but with no constraints on which operating system to use, I'll still lean towards UCSD because of it's convienence.

RA

```
(* BENCHMARK PROGRAM *)
Program matrix;
Const
   maxsize = 45;
   m = 20;
   n = 20;

Var
   i,j,k,l : integer;
   a : array[1..m,1..n] of real;
   b : array[1..n,1..n] of real;
   c : array[1..m,1..n] of real;

   sum : real;
   gup,bell : char;

Procedure filla;
Var
  i,j : integer;

Begin

   for i:=1 to m do
       for j:=1 to n do
           a[i,j]:=i+j;
end;

Procedure fillb;
Var
   i,j : integer;

Begin

   for i:=1 to n do
       for j:=1 to m do
           b[i,j]:=trunc((i+j)/j);
end;

Procedure fillc;
Var
   i,j : integer;

Begin
   for i:= 1 to m do
       for j:=1 to m do
           c[i,j]:=0;
end;
```

```
    Procedure Matmult;
    Var
      i,j,k : integer;

    Begin
       for i:=1 to m do
           for j:=1 to n do
               for k:=1 to m do

                   c[i,j]:=c[i,j] + a[i,k] * b[k,j];
end;

Procedure Summit;
Var
  i,j : integer;

Begin
   for i:=1 to m do
       for j:=1 to m do
           sum:=sum + c[i,j];
end;

Begin {main program}

   write('input any darned number to start it. ')
   readln(gup);
   filla;
   writeln('a filled');
   fillb;
   writeln('b filled');
   fillc;
   writeln('c filled');
   matmult;
   writeln('multiplied');
   summit;
   writeln('sum is : ',sum:8);
   bell:=chr(7);
   writeln(bell);

end.
```

Meeting Door Prize WINNERS

Al Beardsley.....Typing Tutor II
Bob Arvidson .....Beneath Apple DOS

SEED UNCLASSIFIED ADS

Editor: Frank Reeves

WANTED:

8" DISK DRIVES AND/OR CONTROLLER FOR APPLE][, DD/DS OR DD/SS

CALL RON ABBOTT
DAYS 441-3097 or NITES 772-7198

TYPESETTING FROM YOUR OWN COMPUTER
Professional photo typesetting from your word processor text files. Can receive ASCII files via MODEM. For further information call Frank 499-2192

New Pascal Handbook

Book Review by Ron Abbott

Working at a public library I come across many new books for review. A new Pascal tutorial/reference book which I've recently had a chance to peruse is 'The UCSD Pascal Handbook' by Randy Clark and Stephen Koehler. This book is published by Prentice-Hall and retails for $15.95, ISBN # 0-13-935536-7.

While I have several books in my Pascal library, I consider this one the best so far. Why? This book deals with UCSD Pascal, not a generic or ISO version, but the exact one that I use. The book is well written by very knowledgable writers. It is laced with clear understandable examples, and the typeface is most readable. It is has a convenient,easy to read index. I'm always refering to the index it seems, so for me a compleat index is a must!

If I had to have 2 books on Pascal and only 2, I would chose this one and 'Introduction to the UCSD p System' reviewed last month.I consider these two books indespensible.

RA

Minor Epson Interface Problem

by Ron Abbott

Every since I've had my Epson MX-80, it's had one annoying problem, multiple (2) line feeds with every carraige return. I've tried setting the switches every way that seemed correct by to no avail. I finally called Epson tech support in Calif with my problem. The tech rep knew immeadiately what the cause was and he instructed me on the fix.

If you are using an Epson with Apple's Centronics parallel interface card and cable, pin 14 (sec transmit data) on the printer end should be disconnected and allowed to float. If connected this signal is low and resets the dip switch settings to include a linefeed with every carriage return. I disconnected pin 14 and everything worked perfectly.

RA

OCTOBER SIG REPORT - ADVANCED BASIC GROUP

by Dave Eisler

At the October swap meet meeting we reviewed Chapter 3 of the Applesoft Tutorial. Many of the most common commands were discussed and some debugging tricks were demonstrated. We reviewed all the Control and Escape functions and looked at how to change around the text window. To demonstrate various input techniques we made a small data averaging program using several input methods.

Next month we'll probably finish up Chapter 3 and go on to Chapter 4 (GRAPHICS & SOUNDS!). Hope to see you there with your individual questions and problems!

DE

BASIS 108 Introduced

SCOTTS VALLEY, CA - Basis Inc, a subsidiary of 5 year old Basis Microcomputers GmbH of W. Germany introduced to the U.S. market at the Spring Comdex meeting their Apple compatible micro. The system includes a Z80 as well as a 6502 CPU and is also CPM compatible. The list price of the BASIS 108 is $1949. The $419 increased cost buys upper/lower case, 64K RAM, Z80 processor, detached keyboard, numeric keypad, serial & RS 232C 1/o slots design advantages over the Apple which Basis claims is valued at $1200.

Basis also released a 256K ""RAM-DISK'', the Basis 208 Z80 System, and the Basis 216, a 16 brt system supporting either a Z8001 or MC 68000 chip and runs with Microsoft's Xenix Operating system. Basis was an Apple distributer until they refused to renew the contract at the end of 1981. Basis has sold more than 500 units in Europe in the first 6 months of their introduction and 100 were shipped to the U.S. in the first week.

CYCLODS

Review by Vanita Treichel

When I went to the meeting I didn't think I would win. When I looked in the box and all the corners were folded, so I folded mine too. Then they were giving away the door prize winner. I knew it was me because how many people do you know by the name of Vanita.

CYCLODS is a very good game. I did not know how to play and I couldn't find the directions. But I asked my Dad where they were and he found them. Then in about five seconds I was killed. The next time it was better, about ten seconds. The next time it was much better. The game is about this one-eyed guy who you control with the keyboard. You try to smash the snakes with blocks.

There are levels of difficulty from one to twenty. At twenty it is hardest because the snakes move very quickly and the guy moves very quickly also. I still haven't mastered the game because if the snake gets you, you die.

VT

APPLESEEDS OF WISDOM

by Jack D. Quinn

Welcome back to Words of Wisdom for whatever value some of my own personal ideas might be worth. I think it is time to remind everyone of one thing that does horrible things to your Apple II computer not to mention your pocket book for the necessary repairs. If you have not done so, get out your supply of gummed back labels and make one up with the following warning;

"DO NOT REMOVE (OR INSERT) ANY CARD FROM THE APPLE WITH THE POWER TURNED ON".

This also includes disconnecting any wires that my be attached to the plug in cards while the system is powered up. Apple has seen fit to have a warning notice to this effect on the inside of the APPLE III in a place where you can read it when you remove the top cover. They also included a very bright red LED (Light Emmitting Diode) near the card slots to show when the power is on. This might not be a bad idea to have in the Apple II. The usual result of removing a card with the power turned on is a very dead Apple that will not even come on with the APPLE II logo at the top of the screen or boot up a disk. What makes this failure so expensive is that it usually destroys one or more integrated circuits on the main Mother Board and usually one or more chips on each card plugged into the Mother Board slots at the time.

Removing the card itself is not what causes the problem but the way the card is removed. Most people rock the card from front to back and pull up at the same time and this in turn causes the edge fingers on the card to short out the contacts in the Mother Board connector. This sometimes puts some strange voltages on the bus lines and since all the cards are in parallel with each other (including much of the Apple itself) very bad things can happen. (How would you like to be jolted with 12 volts when you were designed to operate with a maximum of 5 volts on one of your pins?) What makes the repair work so expensive is that first you have to make the Apple operate without any cards installed and once you get to that point, put in one card at a time and see if it still operates. If it doesn't then try and find out which integrated circuit is dead and replace it. This can be great fun in itself since many companys solder the integrated circuits into the printed circuit board itself rather than use a 30 cent socket which makes replacement easy. You keep doing this until all of the cards are working. We all have pulled cards out with the power on (I know that I have) so be prepared to have it happen to you.

I see another problem about once a week in our shop that is caused by user error. People get into a hurry and plug the disk drive cable into the Disk II Interface card socket with one row of pins not making contact or else have it so that all the pins are displaced by one row to the right or left in the connector. Doing this results in much smoke comming from the disk drive along with a smell that lingers for days and a repair bill that hurts. What usually happens is the 74LS125 inside the disk drive explodes and usually takes some or all of the other circuits in the drive with it. I also have seen one of the filter capacitors explode. (It looks like a mouse had taken up living inside the disk drive.) There is only one way that works and the Apple DOS book shows how. Pull the interface card out of the Apple and then attach the disk drive rather than try and make the connection inside the Apple. I don't trust myself to get it correct with the card inside the Apple and I do it many times a day. This brings me to the next point: the disk interface card should be plugged into slot 6 of the Apple Mother Board. It will boot up a disk if it is located in another slot but you can get some very strange things to happen if you try a PR#6 and don't have a card there.

Many of you might not realize but there is a sort of standard or convention that has evolved over the years for what cards are put into what slot in the Apple.

This has happened because many software programs require a specific board or card be installed in one of the 8 slots in the Apple to work properly and if there were no standard, a different piece of software would require you to move cards around. Slot 0, which is on the left side of the Apple next to the power supply, is reserved for memory expansion such as a Language card or 16K memory board. An Integer ROM card or an Applesoft ROM card MUST go in this slot. Slot 1 is used for the printer card by most software no mater if it a serial or a parallel card. Slot 2 is used for Modem communications, be it a Hayes Micromodem, Novation Cat, or serial card driving an external Modem. Slot 3 is used for the 80 column video display boards or a serial card driving a terminal. (Yes, you can use a terminal with an Apple!) CP/M and Pascal will come up on a terminal if one is connected to this slot and you can play the game of where did the video go if you have some other card in this slot. Slot 4 is up for grabs, if you have more than 4 disk drives, a disk interface card goes here. Many use the Z80 Microsoft card here. Slot 5 is for a second disk interface card with slot 6 being used for the primary disk drives. Slot 7 can also have the Z80 card or if you are lucky, a 8 inch disk drive controller. All of this information is subject to change and will depend upon YOUR needs but is generally how it is done.

Remember: DON'T PULL CARDS WITH THE POWER ON!!!

JDO

Apple How To - The Apple RPN Calculator

Review by Lloyd Treichel

After attending many monthly meetings it was with wild anticipation that I went home to my Apple and booted the disk that I had received. Most people receive games as door prizes and my kids were very disappointed that all I got was a disk called "Apple How To!" with the subtitle "Lessons in Computing and Calculating". Being an open-minded person I faithfully went through the first recipe displayed which identified this disk having four major parts to it.

To quote the book which accompanies the disk, the "Apple RPN Calculator is a full-blown scientific keystroke calculator with a four-register stack, ten addressable memory registers, and a multiplicity of functions." Somehow the realization that with a piece of software I could emulate a hand-held calculator did not turn me on. I have a couple of hand-held calcualtors and they are very useful. This particular one has none of the advantages of one - the primary one being its lack of portability. On the positive side since all registers of the calculator are visible at all times, it is a good learning tool to understand the workings of a hand-held calculator.

The remainder of the three sections of the disk are tutorials covering a color pattern program, how to use a window in text display, and a section which is a short course in the mini-assembler and the machine language it generates.

This particular disk would be great for the absolute novice on an Apple and would be ideally included with the systems master provided with the system. After having learned much about the Apple through trial and error and reading, these tutorials may have reduced the time it took me to acquire that specific knowledge. In the case of the emulated calculator; having been a programmer for many years and connected with the data processing industry for many more I would have little use for it since I would write a program to resolve the particular issue at hand or if a hand-held calculator were appropriate I would use it.

In a final analysis, I decided not to find out what this particular piece of software was selling for; but would say that I would pay no more than the price of the most inexpensive game available, if I purchased it at all. Someday my children may get into some serious programming and find the disk useful. Right now it has limited value. I would be willing to donate it to a good home or trade it for an inexpensive game.

LT